A Stochastic Optimization Algorithm Using Intelligent Agents

A Program with Constraints and Rate of Convergence

Bao U. Nguyen
Complexity Team

DRDC CORA TM 2010-249
November 2010

Defence R&D Canada
Centre for Operational Research & Analysis

National Defence Défense nationale

Canada

A Stochastic Optimization Algorithm using Intelligent Agents

A Program with Constraints and Rate of Convergence

Bao U. Nguyen
DRDC CORA

Defence R&D Canada – CORA

Technical Memorandum
DRDC CORA TM 2010-249
November 2010

Principal Author

Original signed by Bao U. Nguyen, Ph.D.

Bao U. Nguyen, Ph.D.

Defence Scientist

Approved by

Original signed by D.M. Bergeron, Ph.D.

D.M. Bergeron, Ph.D.

Head Air Section

Approved for release by

Original signed by P. Comeau

P. Comeau

Chief Scientist

Defence R&D Canada – Centre for Operational Research and Analysis (CORA)

Abstract

The problem of optimizing the average time latency of a network, using agents that are able to learn, is examined in this paper. The network design is constrained by a traffic matrix that dedicates specific flows between specific pairs of nodes. Although this is an application type of analysis, only the methodology is presented here, which includes an algorithm for optimization and a corresponding conservative rate of convergence based on no learning. The application part will be presented in the near future once data are available. It is expected that the tools developed in this paper can be used to optimize a wide range of objective functions that do not necessarily have to be the time latency. For example, it could be the cost of the network.

Résumé

Le problème de l’optimisation du temps de latence moyen d'un réseau au moyen d'agents capables d'apprentissage, est examiné dans le présent document. La conception du réseau est contrainte par une matrice de trafic qui établit des flux particuliers entre des paires de nœuds particulières. Bien qu'il s'agisse d'un type de mise en application d'analyse, seulement les méthodologies sont présentées ici, y compris un algorithme d'optimisation et un taux de convergence correspondant raisonnable fondés sur un modèle sans apprentissage. La mise en application sera présentée prochainement, une fois les données disponibles. Il est espéré que les outils décrits dans le présent document permettront d'optimiser une vaste gamme de fonctions objectifs en plus de la latence.

This page intentionally left blank.

Executive summary

A Stochastic Optimization Algorithm using Intelligent Agents: With Constraints and Rate of Convergence

Bao U. Nguyen; DRDC CORA TM 2010-249; Defence R&D Canada – CORA; November 2010.

Background: The power of computers nowadays allows us to examine problems where close form solutions do not necessarily exist, but which can be solved using efficient algorithms. In addition, the models based on algorithms can simulate a level of detail that close form solutions often cannot. In this Technical Memorandum (TM), one such algorithm is used to model a network such as the one implemented in the Networked Underwater Warfare Technology Demonstration Program (Ref [1]) that was conducted at DRDC Atlantic.

Results: The current algorithm consists of intelligent agents who can learn. The learning process of the agents leads to optimization of an objective function that is subject to a number of constraints. The objective function was chosen to be the average time latency of a network, and the constraints to be the traffic matrix. The aim is to minimize the average time latency while maintaining dedicated flows among pairs of nodes that form a network. A node can be a sensor, a ship, an aircraft, a submarine, etc. However, this algorithm is general in the sense that it can optimize other objective functions that are not the time latency and can model other types of constraints that are not the traffic matrix.

Significance: This report describes a novel optimization algorithm as well as the rate of convergence for the algorithm. This is a new theoretical development for heuristic algorithms that simulate Markov processes. It gives the Operations Research practitioner a valuable tool to determine how good an algorithm is and how optimal the solution is. To the best of the author's knowledge, such a convergence criteria is not available in the open literature. It is hoped that the reader will make use of this type of agent-based algorithm and the corresponding convergence rates in the application of heuristic optimization algorithms.

Sommaire

A Stochastic Optimization Algorithm using Intelligent Agents: With Constraints and Rate of Convergence

Bao U. Nguyen; DRDC CORA TM 2010-249; R & D pour la défense Canada – CARO; Novembre 2010.

Contexte: La puissance des ordinateurs nous permet aujourd'hui d'étudier des problèmes pour lesquels une solution analytique n'existe peut-être pas, mais qui peuvent être résolus au moyen d'algorithmes efficaces. De plus, les modèles fondés sur les algorithmes peuvent simuler un niveau de détail que les solutions analytiques ne peuvent pas toujours atteindre. Le présent document technique (TM) décrit un algorithme de ce genre utilisé pour modéliser un réseau tel que celui mis en œuvre dans le projet de démonstration de technologies sur la guerre sous-marine en réseau (Ref [1]) qui avait été réalisé à RDDC Atlantique.

Résultats: Le présent algorithme est constitué d'agents intelligents capables d'apprentissage. Le processus d'apprentissage des agents permet d'optimiser une fonction objectif assujettie à un certain nombre de contraintes. La fonction objectif choisie est le temps de latence moyen d'un réseau et les contraintes sont obtenues de la matrice de trafic. Autrement dit, l'objectif est de minimiser le temps de latence moyen tout en maintenant les flux entre les paires de nœuds qui forment le réseau. Un nœud peut être, par exemple, un capteur, un navire, un aéronef, un sous-marin. Il n'en reste pas moins que l'algorithme est général : il peut être utilisé pour optimiser des fonctions objectifs autres que la latence et pour modéliser des types de contraintes autres que la matrice de trafic

Importance: Ce rapport décrit non seulement un algorithme d'optimisation novateur, mais aussi le taux de convergence de l'algorithme. Il s'agit d'un nouveau développement théorique des algorithmes heuristiques qui simulent les processus de Markov. Le praticien en recherche opérationnelle obtient ainsi un précieux outil pour évaluer la qualité de son algorithme et pour déterminer si sa solution est optimale. À la connaissance de l'auteur, aucun autre critère de convergence similaire n'est encore disponible dans les sources publiées. Il est espéré que le lecteur mettra à profit ce type d'algorithme fondé sur les agents et les taux de convergence correspondants dans la mise en application d'algorithmes d'optimisation heuristique.

Table of contents

List of figures

Acknowledgements

The author wishes to thank Kevin Ng who brought to his attention the paper of Oommen and Roberts 2000.

This page intentionally left blank.

1 Background

In support of a Technology Investment Fund (TIF) project at DRDC CORA on Network Centric Warfare (Ref [2]), the time latency of a generic network that is subject to a number of constraints, is determined. The time latency of a network can be determined in many ways; however, this paper will describe a methodology on how to optimize the time latency using agents that have the ability to learn. The main results of this paper are to provide an algorithm to do so and to derive the rate of convergence of the corresponding algorithm when no learning is in effect. The methodology is inspired from Ref [3], which provides a heuristic algorithm to optimize a cost objective function. Although part of the material in this Technical Memorandum (TM) has been published in Ref [4], the report includes an improvement in the convergence rate of the optimization algorithm and a more complete proof of this convergence rate.

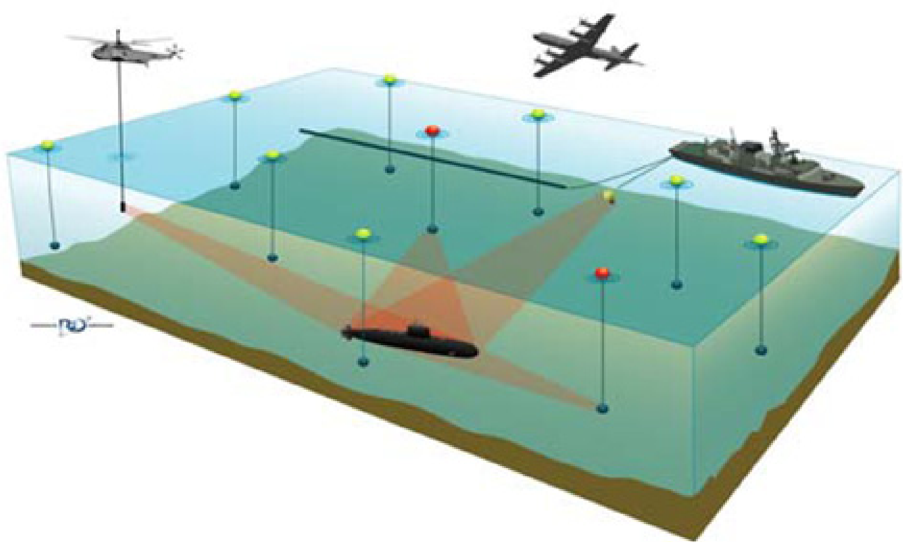

Figure 1: An Example of a Defence Network.

2 Learning Algorithm

This section examines a communication network that has a globally maximal capacity C. The network flows must satisfy the traffic matrix (γ_{uv}). That is, there will be a dedicated flow from node u to node v that is greater than or equal to γ_{uv}. This ensures that node u can communicate with node v with the desired flow γ_{uv}. In addition, the time latency of the network depends on both the flow and the capacity of each link. This development makes use of the algorithm in Ref [3], which is described below.

The capacities of the links are represented by a vector $(c_1, c_2, ..., c_e)$ where e is the number of edges or links in the network, c_i is chosen from a finite set of capacities such as $(0,1,2,...)$ where a capacity unit corresponds, for example, to $1200\, bps$ (bits per second). For each link i and each possible capacity j, there is a triplet (I_{ij}, S_{ij}, D_{ij}) where I_{ij} is the probability that the current capacity j of the link i should be increased, S_{ij} is the probability that the current capacity j of the link i should remain unchanged, and D_{ij} is the probability that the current capacity j of the link i should be decreased. The capacity c_i of a link i is modelled as an agent whose learning process is encoded in the evolution of the triplet (I_{ij}, S_{ij}, D_{ij}) where $I_{ij} + S_{ij} + D_{ij} = 1$.

The final solution vector will consist of the capacities c_i such that S_{ij} probability values approach unity e.g. 0.99. The closer this value is to unity, the more accurate is the solution. This is so that as the optimal solution is approached, the algorithm favours the current solution, hence S_{ij} tends to unity. As is often the case with heuristic algorithms, Ref [3] did not provide the rate of convergence of their algorithm. Fortunately, it is possible to derive the rate of convergence for this algorithm, at least in the case where agents do not learn. However, before presenting the derivation for the rate of convergence, the pseudo-code in Ref [3] is shown below. The algorithm can be divided into three modules. Module 1 initializes the triplet (I_{ij}, S_{ij}, D_{ij}), looks for a feasible solution, and determines its value as dictated by the objective function. Module 2 searches the solution space. Module 3 updates the triplet (I_{ij}, S_{ij}, D_{ij}).

Module 1. Initialize the triplet (I_{ij}, S_{ij}, D_{ij}).

For ($i = 1$ to $\text{maxlinks}(=\text{e})$)
    For ($j = 1$ to $\text{maxcaps}(=\text{C})$)
        If ($j = 1$ (left-boundary-state))
            $I_{ij} = 1/2$, $S_{ij} = 1/2$, $D_{ij} = 0$
        End-If
        If ($j = \text{maxcaps}$ (right-boundary-state))
            $I_{ij} = 0$, $S_{ij} = 1/2$, $D_{ij} = 1/2$
        End-If
        If ($1 < j < C$ (internal-state))
            $a = 1/3$, $I_{ij} = a$, $S_{ij} = a$, $D_{ij} = a$
        End-If
    End-For
End-For
Repeat
    For ($i = 1$ to maxlinks)

$c_i = RAND(0, \text{maxcaps})$

End-For

Until (network is feasible)

current-objective = calculate-objective()

For ($i = 1$ to maxlinks)

$best - c_i = c_i$

End-For

best-objective = current-objective()

Module 2. Search the solution space.

While (count<num-iterations) and (accuracy-level (all links) < required accuracy)

For ($i = 1$ to maxlinks)

$Action_i = RAND(Increase_{ij}, Stay_{ij}, Decrease_{ij})$

If ($Action_i = Increase_{ij}$)

$c_i = c_i + 1$

End-If

If ($Action_i = Decrease_{ij}$)

$c_i = c_i - 1$

End-If

current-objective = calculate-objective()

End-For

For ($i = 1$ to maxlinks)

$j = c_i$

If (network is feasible)

If ($Action_i = Increase_{ij}$)

$\boldsymbol{Raise}(D_{ij}, \lambda_{R1})$

End-If

If ($Action_i = Stay_{ij}$)

$\boldsymbol{Raise}(S_{ij}, \lambda_{R1})$

End-If

If ($Action_i = Decrease_{ij}$)

$\boldsymbol{Raise}(D_{ij}, \lambda_{R1})$

End-If

Else

Reset all links to best-objective capacities

End-If

If (network is feasible) and (current-objective < best-objective)

If ($Action_i = Increase_{ij}$)

$\boldsymbol{Raise}(D_{ij}, \lambda_{R2})$

End-If

If ($Action_i = Stay_{ij}$)

$\boldsymbol{Raise}(S_{ij}, \lambda_{R2})$

End-If

If ($Action_i = Decrease_{ij}$)

$\boldsymbol{Raise}(S_{ij}, \lambda_{R2})$

End-If

For ($i = 1$ to maxlinks)

$best - c_i = c_i$

End-For

best-objective = current-objective()

End-If

End-For

End-While

Module 3. Procedure Raise. Updating the triplet (I_{ij}, S_{ij}, D_{ij}). $\lambda_R = \lambda_{R1}$ is associated with a new feasible solution. $\lambda_R = \lambda_{R2}$ is associated with a new feasible solution that is also superior.

If (Action = Increase)

$D_{ij} = \lambda_R \cdot D_{ij}$; $S_{ij} = \lambda_R \cdot S_{ij}$; $I_{ij} = 1 - (D_{ij} + S_{ij})$

End-If

If (Action = Stay)

$I_{ij} = \lambda_R \cdot I_{ij}$; $D_{ij} = \lambda_R \cdot D_{ij}$; $S_{ij} = 1 - (I_{ij} + D_{ij})$

End-If

If (Action = Decrease)

$I_{ij} = \lambda_R \cdot I_{ij}$; $S_{ij} = \lambda_R \cdot S_{ij}$; $D_{ij} = 1 - (I_{ij} + S_{ij})$

End-If

3 Algorithm Extension

The capacity assignment is modelled in the same way as that of Oommen and Roberts 2000 (see previous section). Hence, there is a triplet $\left(I_{ij}^{(c)}, S_{ij}^{(c)}, D_{ij}^{(c)}\right)$ associated with link i and capacity j, and the superscript c stands for capacity. In addition, each path l is modelled in a similar way as an agent that carries k units of flow and which connects node u to node v. Each path type agent is represented by a triplet $\left(I_{uvlk}^{(p)}, S_{uvlk}^{(p)}, D_{uvlk}^{(p)}\right)$ where the superscript p stands for path. These triplets are updated at each run depending on random numbers and whether the objective function is improved or not. For example, if the objective function is improved then both $S_{ij}^{(c)}$ and $S_{uvlk}^{(p)}$ are increased. This new algorithm is the first main result of this paper and can be used to optimize the average time latency of a network, as defined in Ref [5]:

$$T = \frac{1}{\sum_{uv} \gamma_{uv}} \cdot \sum_k \frac{f_k}{c_k - f_k} \tag{1}$$

where (u,v) represents node u and node v; each k represents a link while f_k and c_k are respectively the flow and capacity of that link. Enumeration is used to generate all possible paths that connect node u to node v. Modelling each path as an agent ensures flow conservation through each node. The flow through a link is then equal to the sum of the flows of all paths that traverse that link. For example, let $\{a,b,c,d,e\}$ be the set of nodes of a complete graph (all possible links) as shown in Figure 2. Let's consider the link $(a-b)$; path1 be $(a-b-c)$ with 2 units of flow, path2 be $(a-b-d)$ with 1 unit of flow and path3 be $(c-a-b)$ with 3 units of flow. The flow through the link $(a-b)$ will be the sum of $2+1+3$ as each of the three paths traverse $(a-b)$. Note that the flow of a link ranges from zero to C (the maximal capacity of the network).

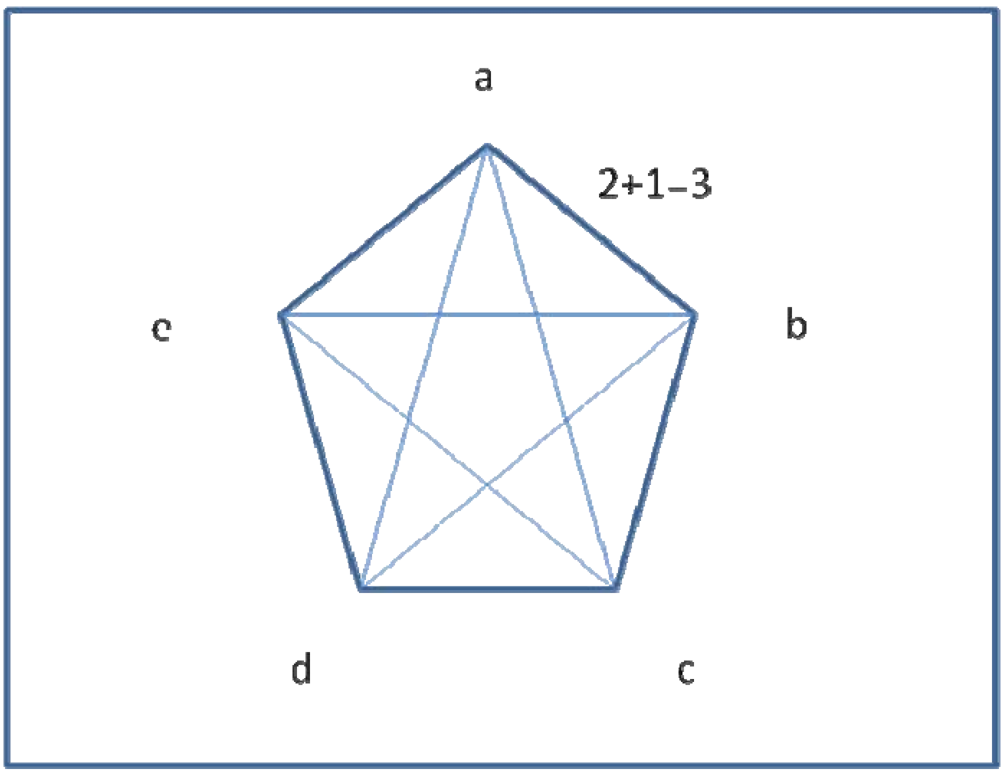

Figure 2: An Example of Flows in a Graph.

4 Rate of Convergence of Flows in a Graph

Define the rate of convergence in a similar way to that in Ref [6]. That is, the probability that a globally optimal state is found at least once. The rate of convergence derived below assumes no learning. Since the purpose of learning is to accelerate the convergence of the algorithm, it is expected that this rate of convergence is a conservative estimate of the algorithm. Observe that for a link, the flow through that link is regulated by a Markov chain as shown in Fig. 3. Each flow state is labelled by a number. For example, the label zero indicates that the flow is equal to zero (unit of flow) while the label one indicates that the flow is equal to one (unit of flow) etc. The connections between the labels show the transitions among the states. For example, the link from flow zero to flow one represents the probability that flow zero makes a transition to flow one. The link from flow one to flow one represents the probability that flow one remains flow one.

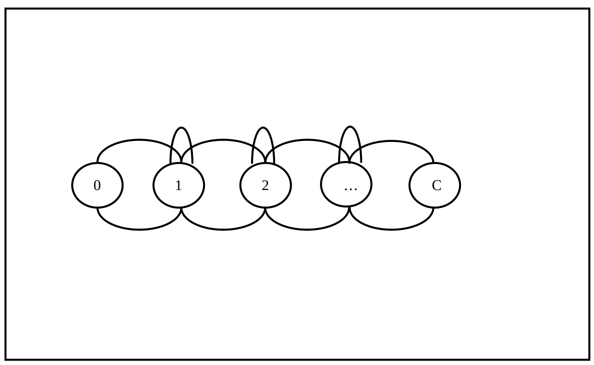

Figure 3: A Markov Chain.

The Markov chain shown in Fig. 3 can be represented by a transition matrix P where P_{ij} is the probability that state i transitions into state j. For example, given $C = 4$, we get:

$$P = \begin{array}{c c} & \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ C=4 \end{array} & \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ a & a & a & 0 & 0 \\ 0 & a & a & a & 0 \\ 0 & 0 & a & a & a \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array} \tag{2}$$

where $a = 1/3$. Model P that way so that the state with flows equal to zero or C are not considered feasible. If a flow of a link is equal to zero, then there is no communication necessary. Hence, the probability of transition from state zero to state one is set to 100 percent. If a flow of a link is equal to C, then there is no capacity left for the remaining links, which can happen only when there is only one link in the network. However, the network considered here consists of many links. Hence, the probability of transition from state C to state $C-1$ is also set to 100 percent. Additionally, for $j=1,...,C-1$, given link i, the probability, $P_{j,j-1}$, that state j transitions to state $j-1$ is equal to $D_{i,j}=a$; the probability, $P_{j,j}$, that state j stays the same is equal to $S_{i,j}=a$; and the probability, $P_{j,j+1}$, that state j transitions to state $j+1$ is equal to $I_{i,j}=a$. For example, $P_{1,0}=P_{1,1}=P_{1,2}=a=D_{i,1}=S_{i,1}=I_{i,1}$ while all other transitions from state 1 are forbidden.

4.1 Lemma 1

The probability that the flow through a path is optimal is given by:

$$p_f \geq 1-(1-a)\cdot\frac{(C-2)}{(C-1)^2}\cdot\left|Q_f^n\right| \tag{3}$$

where $a=1/3$, n is the number of iterations, P_f is the transition matrix associated with the flow of a path; Q_f is equal to P_f with the exceptions that the first row, first column, last row and last column elements are set to zero; and $\left|Q_f{}^n\right|$ is the sum of all elements of $Q_f{}^n$. That is, $\left|Q_f{}^n\right|=\sum_i\sum_j\left(Q_f{}^n\right)_{i,j}$

For example, given $C=4$, we get:

$$Q_f=\begin{pmatrix}0&0&0&0&0\\0&a&a&0&0\\0&a&a&a&0\\0&0&a&a&0\\0&0&0&0&0\end{pmatrix}$$

$$\left|Q_f\right|=7\cdot a \tag{4}$$

Note that, P_f and Q_f do not change with n. This is so because there is no learning. However, when learning is in effect, the elements of P_f and Q_f are updated through the triplets (I,S,D).

4.2 Proof of Lemma 1

To alleviate the notation, suppress the subscript f associated with P and Q. Let the optimal flow be f^*. We wish to determine the probability that a random and non–optimal flow i transitions to another, also random and non–optimal, flow j. The probability of picking a random flow such that $i \neq f^*$ is:

$$P\left(flow = i \neq f^* \right) = \frac{(C-2)}{(C-1)^2}$$

For example, given $C = 4$, then there are five possible states belonging to $\{0,1,2,3,4\}$. A random flow can be equal to any of these five states. However, if state zero and state four are discarded as non feasible states, there remain only three states. Hence the probability of picking a random and feasible flow i is $\frac{1}{(C-1)} = 1/3$. Further, let $f^* = 3$, then the probability that a random and feasible flow is not optimal is $\frac{(C-2)}{(C-1)} = 2/3$. As a result, the probability of picking at random a flow i and that flow i is feasible and not optimal is $\frac{(C-2)}{(C-1)^2} = (1/3)\cdot(2/3) = 2/9$.

The probability of starting with a feasible flow $i \neq f^*$ and ending up with another feasible flow $j \neq f^*$, is regulated by the transition matrix P. That is,

$$P\left(i \neq f^* \to j \neq f^*\right) \leq P\left(i \neq f^*\right) \cdot P_{i,j} \cdot (1-a) = (1-a) \cdot \frac{(C-2)}{(C-1)^2} \cdot P_{i,j} \tag{5}$$

The probability of starting with any feasible flow $i \neq f^*$ and ending up with any other feasible flow $j \neq f^*$ is the sum of the above expression over i and j such that $i, j \neq f^*$ i.e.

$$(1-a) \cdot \frac{(C-2)}{(C-1)^2} \cdot \sum_{i,j \neq f^*} P_{ij}$$

Furthermore, observe that

$$\sum_{i,j \neq f^*} P_{i,j} \leq \sum_{i,j} P_{i,j}$$

since all the elements of the transition matrix P are non negative and the LHS sums over all elements i and j such that $i, j \neq f^*$ while the RHS sums over all elements i and j with no restrictions. That is, the sum on the RHS includes more elements of P than the sum on the LHS. Therefore,

$$(1-a)\cdot\frac{(C-2)}{(C-1)^2}\cdot\sum_{i,j\neq f^*}P_{i,j}\leq(1-a)\cdot\frac{(C-2)}{(C-1)^2}\cdot\sum_{i,j}P_{i,j}$$

Additionally, the RHS, above, sums over the states that are feasible and non–optimal, which implies that $i, j \neq 0, C$. This can be interpreted in a way such that the RHS above includes all transitions from a state i that is feasible to a state j that is also feasible. Therefore, any transitions from (to) 0 or C to (from) a feasible state is forbidden. This is equivalent to replace P_{ij} by Q_{ij}. If this argument is repeated n times, we get an upper bound for the probability q_f of not achieving the optimal state after n iterations:

$$q_f\leq(1-a)\cdot\frac{(C-2)}{(C-1)^2}\cdot\sum_{i,j}Q^n{}_{i,j}=(1-a)\cdot\frac{(C-2)}{(C-1)^2}\cdot\left|Q^n\right|$$

Therefore, the lower bound to the probability $p_f = 1-q_f$ of achieving the optimal state satisfies:

$$p_f\geq1-(1-a)\cdot\frac{(C-2)}{(C-1)^2}\cdot\left|Q^n\right|$$

4.3 Lemma 2

The probabilistic bound in Eqn. (3) of finding the optimal flow is an increasing function of n.

4.4 Proof of Lemma 2

This is true as

$$Q^{n+1}=Q^n\cdot Q=Q^n\cdot(P-\Delta)$$

where $\Delta = P - Q$. Simple algebra dictates that

$$\left|Q^n\cdot P\right|=\left|Q^n\right|$$

Hence,

$$\left|Q^{n+1}\right|=\left|Q^n\cdot P\right|-\left|Q^n\cdot\Delta\right|=\left|Q^n\right|-\left|Q^n\cdot\Delta\right|$$

Since $Q^n \cdot \Delta$ has only positive and zero elements, this means that $\left|Q^n \cdot \Delta\right| > 0$. Therefore,

$$\left|Q^{n+1}\right| < \left|Q^n\right|$$

As

$$p_f \geq 1-(C-2)/(C-1)^2 \cdot \left|Q^n\right|$$

This implies that p_f is an increasing function of n. For example, given $C=4$,

$$\left|Q \cdot \Delta\right| = 4 \cdot a^2$$

Hence,

$$\left|Q^2\right| = \left|Q\right| - 4 \cdot a^2 < \left|Q\right|$$

4.5 Lemma 3

The probability that the capacity of a link is optimal is given by

$$p_c \geq \min_{c=1,\ldots C}\left\{1-(1-a)\cdot\frac{(c-2)}{(c-1)^2}\cdot\left|Q_c^n\right|\right\} = 1-(1-a)\cdot\frac{(C-2)}{(C-1)^2}\cdot\left|Q_C^n\right| \quad (6)$$

Note that p_c is the probability of achieving the optimal capacity and $Q_C = Q_f$. However, if we assume that learning occurs, then Q_f and Q_C will change as a function of n, in which case they will not necessarily evolve in the same way. Observe that the capacity of a link must be greater than the flow through that link since otherwise the average time latency shown in Eqn. (1) is ill defined. If a flow is j, then the capacity ranges from $j+1$ to C. If the capacity shifts to the left by j, we get c ranging from one to $C-j$. Since $j \geq 0$, the largest value for c is C. The fact that the largest value of c is C and that

$$\frac{(c-1)}{c^2}\cdot\left|Q_{c+1}^n\right| \geq \frac{(c-2)}{(c-1)^2}\cdot\left|Q_c^n\right|$$

for all c, as proved below, allows us to assert Lemma 3.

4.6 Proof of Lemma 3

Proceed by induction on n. The first step is to prove that the lemma holds for $n=1$. That is,

$$\frac{(c-1)}{c^2}\cdot|Q_{c+1}|\geq\frac{(c-2)}{(c-1)^2}\cdot|Q_c|$$

Since $\frac{(c-1)}{(c)}\geq\frac{(c-2)}{(c-1)}$, we only need to show that $\frac{1}{c}\cdot|Q_{c+1}|\geq\frac{1}{(c-1)}\cdot|Q_c|$. This can be established by observing that $|Q_c|=(c-3+4/3)$.

Now, assume that this is true for all $k=1,\ldots,n$ and prove for $n+1$. We will show that this is true when c is odd. A similar proof can be shown when c is even.

Let $Q_c^n=\left[\vec{d}_1^{\,n};\vec{d}_2^{\,n};\ldots;\vec{d}_{c-1}^{\,n}\right]$ where $\vec{d}_i^{\,n}$ is the *ith* column of Q_c^n and $Q_{c+1}^n=\left[\vec{e}_1^{\,n};\vec{e}_2^{\,n};\ldots;\vec{e}_c^{\,n}\right]$ where $\vec{e}_i^{\,n}$ is the *ith* column of Q_{c+1}^n. Note that the boundary rows and columns of Q_c^n and Q_{c+1}^n, whose elements are zeroes, were removed. This will not affect the proof. Q_c^n obeys a recursion:

$$Q_c^n\cdot Q_c=a\cdot\left[\vec{d}_1^{\,n}+\vec{d}_2^{\,n};\vec{d}_1^{\,n}+\vec{d}_2^{\,n}+\vec{d}_3^{\,n};\vec{d}_2^{\,n}+\vec{d}_3^{\,n}+\vec{d}_4^{\,n};\ldots;\vec{d}_{c-2}^{\,n}+\vec{d}_{c-1}^{\,n}\right] \quad (7)$$

Q_{c+1}^n obeys a similar recursion to the one above.

4.6.1 Case 1: Assume that c is odd

Applying recursion and induction repeatedly, we get p inequalities where $p=\left\lfloor\frac{c}{2}\right\rfloor$. Each time, the new inequality is obtained by removing the first term and the last term on both the LHS and RHS of the previous inequality.

$$\frac{1}{c}\cdot\left(\left|\vec{e}_2^{\,n-1}\right|+\ldots+\left|\vec{e}_{c-1}^{\,n-1}\right|\right)\geq\frac{1}{c-1}\cdot\left(\left|\vec{d}_2^{\,n-1}\right|+\ldots+\left|\vec{d}_{c-2}^{\,n-1}\right|\right)$$
$$\frac{1}{c}\cdot\left(\left|\vec{e}_3^{\,n-2}\right|+\ldots+\left|\vec{e}_{c-2}^{\,n-2}\right|\right)\geq\frac{1}{c-1}\cdot\left(\left|\vec{d}_3^{\,n-2}\right|+\ldots+\left|\vec{d}_{c-3}^{\,n-2}\right|\right)$$
...

$$\begin{aligned}&\frac{1}{c}\cdot\left(\left|\vec{e}_{p}^{\,n-p}\right|+\left|\vec{e}_{p+1}^{\,n-p}\right|+\left|\vec{e}_{p+2}^{\,n-p}\right|\right)\\&\geq\frac{1}{c-1}\cdot\left(\left|\vec{d}_{p}^{\,n-p}\right|+\left|\vec{d}_{p+1}^{\,n-p}\right|+\left|\vec{d}_{p+2}^{\,n-p}\right|\right)\end{aligned}\tag{8}$$

$$\begin{aligned}&\frac{1}{c}\cdot\left(\left|\vec{e}_{p}^{\,n-p-1}\right|+2\cdot\left|\vec{e}_{p+1}^{\,n-p-1}\right|+\left|\vec{e}_{p+2}^{\,n-p-1}\right|\right)\\&\geq\frac{1}{c-1}\cdot\left(\left|\vec{d}_{p}^{\,n-p-1}\right|+\left|\vec{d}_{p+1}^{\,n-p-1}\right|\right)\end{aligned}\tag{9}$$

where $\left|\vec{x}\right|$ is the sum of all elements of the vector $\vec{x}$. The lemma is true if the first inequality is true. But the first inequality is true if the second inequality is true. Repeating the argument tells us that the lemma is true if the last inequality is true. Again, by induction, Eqn. (8) is true when replacing $n-p$ by $n-p-1$:

$$\begin{aligned}&\frac{1}{c}\cdot\left(\left|\vec{e}_{p}^{\,n-p-1}\right|+\left|\vec{e}_{p+1}^{\,n-p-1}\right|+\left|\vec{e}_{p+2}^{\,n-p-1}\right|\right)\\&\geq\frac{1}{c-1}\cdot\left(\left|\vec{d}_{p}^{\,n-p-1}\right|+\left|\vec{d}_{p+1}^{\,n-p-1}\right|+\left|\vec{d}_{p+2}^{\,n-p-1}\right|\right)\end{aligned}\tag{10}$$

Combining Eqns. (9) and (10), we get:

$$\begin{aligned}&\frac{1}{c}\cdot\left(\left|\vec{e}_{p}^{\,n-p-1}\right|+2\cdot\left|\vec{e}_{p+1}^{\,n-p-1}\right|+\left|\vec{e}_{p+2}^{\,n-p-1}\right|\right)\\&\geq\frac{1}{c}\cdot\left(\left|\vec{e}_{p}^{\,n-p-1}\right|+\left|\vec{e}_{p+1}^{\,n-p-1}\right|+\left|\vec{e}_{p+2}^{\,n-p-1}\right|\right)\\&\geq\frac{1}{c-1}\cdot\left(\left|\vec{d}_{p}^{\,n-p-1}\right|+\left|\vec{d}_{p+1}^{\,n-p-1}\right|+\left|\vec{d}_{p+2}^{\,n-p-1}\right|\right)\end{aligned}$$

Hence the proof is complete because we have shown that the inequality (9) is true.

4.6.2 Case 2: Assume that c is even

Applying recursion and induction to $\frac{\left|Q_{c+1}^{n}\right|}{c}\geq\frac{\left|Q_{c}^{n}\right|}{c-1}$ to get p inequalities where $p=\frac{c}{2}$. Use the same technique as in case 1.

$$
\begin{aligned}
&\frac{1}{c}\cdot\left(\left|\overline{e}_2^{\,n-1}\right|+\ldots+\left|\overline{e}_{c-1}^{\,n-1}\right|\right)\geq\frac{1}{c-1}\cdot\left(\left|\overline{d}_2^{\,n-1}\right|+\ldots+\left|\overline{d}_{c-2}^{\,n-1}\right|\right)\\
&\frac{1}{c}\cdot\left(\left|\overline{e}_3^{\,n-2}\right|+\ldots+\left|\overline{e}_{c-2}^{\,n-2}\right|\right)\geq\frac{1}{c-1}\cdot\left(\left|\overline{d}_3^{\,n-2}\right|+\ldots+\left|\overline{d}_{c-3}^{\,n-12}\right|\right)\\
&\ldots\\
&\frac{1}{c}\cdot\left(\left|\overline{e}_p^{\,n-p}\right|+\left|\overline{e}_{p+1}^{\,n-p}\right|\right)\geq\frac{1}{c-1}\cdot\left(\left|\overline{d}_p^{\,n-p}\right|\right)
\end{aligned}
\tag{11}
$$

The inequalities above are true if and only if:

$$
\frac{1}{c}\cdot\left(\left|\overline{e}_p^{\,n-p-1}\right|+\left|\overline{e}_{p+1}^{\,n-p-1}\right|\right)\geq 0
$$

But the above is true as all elements of the matrices Q_c^n are non negative. Hence the proof is complete.

4.7 Corollary of Lemma 3

Combining the result of Lemma 1 to that of Lemma 3, we obtain the second main result of this paper, the lower bound to the probability of finding the optimal solution in a network that has e links and s paths between all pairs of nodes:

$$
\begin{aligned}
p &\geq\left(1-(1-a)\cdot\frac{(C-2)}{(C-1)^2}\cdot\left|Q_C^n\right|\right)^s\cdot\left(1-(1-a)\cdot\frac{(C-2)}{(C-1)^2}\cdot\left|Q_C^n\right|\right)^e\\
&=\left(1-(1-a)\cdot\frac{(C-2)}{(C-1)^2}\cdot\left|Q_C^n\right|\right)^{e+s}
\end{aligned}
\tag{12}
$$

4.8 Proof of Corollary

The probability of finding the optimal solution is the product of the probability that each path carries the optimal flow (provided by Eqn. (3)) and the probability that each link has the optimal capacity (provided by Eqn. (6)). That is, following the inequality sign in Eqn. (12), the first factor is the lower probabilistic bound of finding the optimal flows where s is the total number of paths between all pairs of nodes, while the second factor is the lower probabilistic bound of finding the optimal capacities for e links.

4.9 Example

Figure 4 below shows that the probability of achieving the optimal solution increases as a function of number of runs. This is so as the larger the number of runs, the higher the probability of achieving the optimal solution. $C = 20$ $(C = 21 \text{ and } C = 22)$ means the flow through each link ranges from zero to twenty (zero to twenty one and zero to twenty two). Note that as C decreases, the search space decreases and hence it is easier to find the optimal solution and therefore the probability of achieving the optimal solution increases. Figure. 4 assumes the network shown in Figure 2, i.e., there are ten links $(e = 10)$ and ten pairs of nodes. Assuming five paths per pair of nodes yields the parameter $s = 50 = 10 \cdot 5$. Based on Eqn. (12), the lower bound for the probability of finding the optimal solution is:

$$p \geq \left(1 - (1-a) \cdot \frac{(C-2)}{(C-1)^2} \cdot \left|Q_C^n\right|\right)^{60}$$

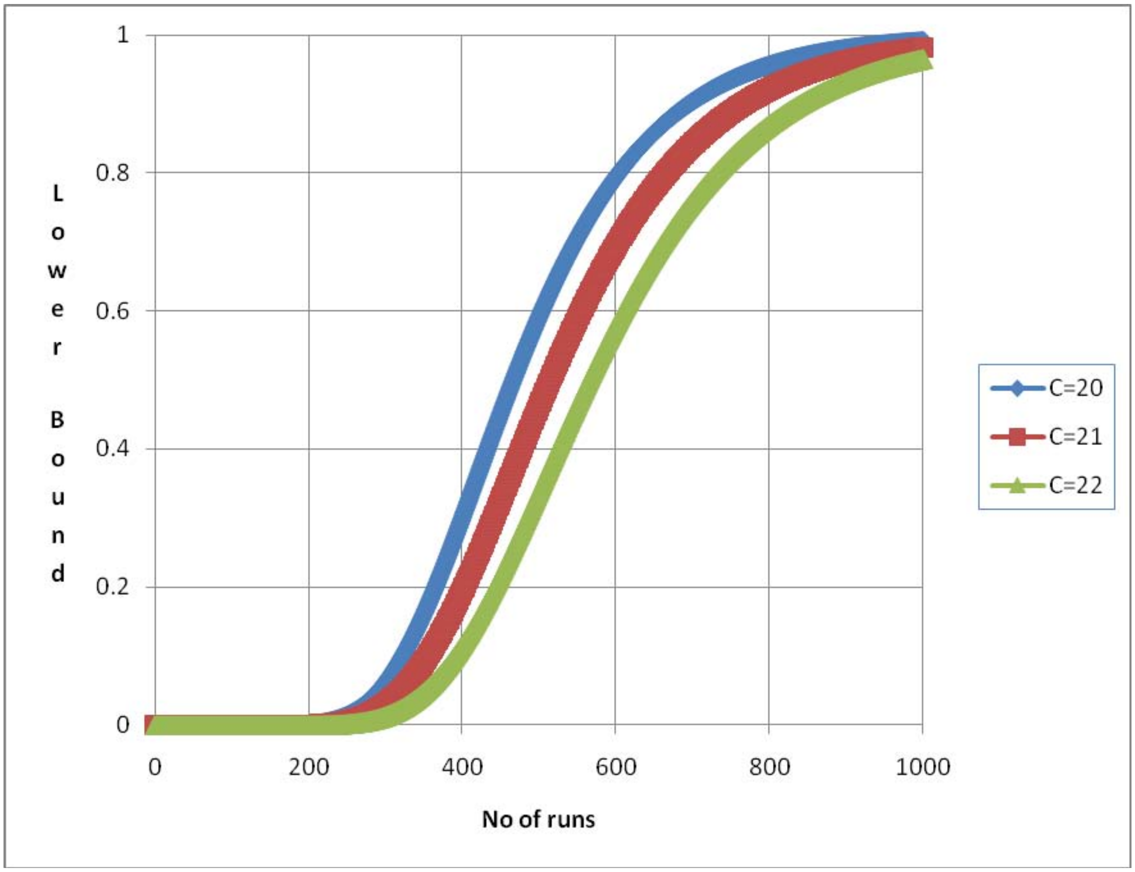

Figure 4: Lower Bound to the Probability of Achieving the Optimal Solution as a Function of Number of Runs.

4.10 Discussion

The reader might wonder what happens when a is equal to zero. If it was the case, then

$$p \geq \left(1-(1-a)\cdot\frac{(C-2)}{(C-1)^2}\cdot\left|Q_C^n\right|\right)^{60} = 1$$

as

$$Q_C^n = 0$$

hence

$$\left|Q_C^n\right| = 0$$

for all n, which implies that we would find the optimal state at the first iteration, i.e., when n is equal to one. This seems paradoxical. The answer lies in the fact that n is the number of iterations that will produce feasible solutions. As a result, when $Q_C = 0$, there is no feasible solution because the probability of transition from a feasible state to another feasible state as dictated by the matrix Q_C is zero.

Even though, in this report, a specific type of Markov matrix Q_C (as shown in Eqn(4)) was chosen, it is believed that the methodology developed here can be extended to a more general class of Markov processes. For example, transitions do not necessarily have to be among nearest neighbours. State 1 can transition to state 3 without going through state 2. This means Q_C can be more general as shown below:

$$Q_C = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & a_{1,1} & a_{1,2} & a_{1,3} & 0 \\ 0 & a_{2,1} & a_{2,2} & a_{2,3} & 0 \\ 0 & a_{3,1} & a_{3,2} & a_{3,3} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{13}$$

5 Conclusion

This paper described a new agent-based algorithm that optimizes an objective function depending on both the flow and the capacity of each link, and that satisfies the traffic matrix constraint. In addition, a novel rate of convergence was derived for this algorithm when assuming no reinforcement learning. It is believed that the rate of convergence, when reinforcement learning is imposed, can be further derived.

References

[1] Le François, M. 2004. “Networked Underwater Warfare”, Maritime Warfare Bulletin, Canadian Forces Maritime Warfare Centre, Issue 2004, pp 66-68.

[2] Ng, Kevin, Roy Mitchell and Bao Nguyen. “Networked enabled operations and networked enabled capability performance metrics”, Technology Investment Fund proposal, May 2006.

[3] Oommen, B. J. and T. Dale Roberts. 2006. “Continuous Learning Automata Solutions to the Capacity Assignment Problem”, IEEE Transactions on computers, Vol 49, No 6, pp 608-620.

[4] Nguyen, B. U. 2009. “A new stochastic algorithm for the capacity & flow assignment problem and an associated rate of convergence”, 23rd European Conference on Modelling and Simulation, Madrid, Spain, Jun 09-12, 2009.

[5] Mostafa, M. E. and Saad M. A. Eid. 2000. “A genetic algorithm for joint optimization of capacity and flow assignment in packet switched networks”, 17th national radio science conference, Minufiya University, Egypt, pp C5.1-C5.6.

[6] Rajasekaran, Sanguthevar 1990 “On the convergence time of simulated annealing”, University of Pennsylvania, Department of Computer & Information Science Technical Report (CIS).

List of symbols/abbreviations/acronyms/initialisms

CORA	Centre for Operational Research and Analysis
DND	Department of National Defence
DRDC	Defence Research & Development Canada
DRDKIM	Director Research and Development Knowledge and Information Management
OR	Operations Research
R&D	Research & Development
TM	Technical Memorandum

Glossary

Technical term	Explanation of term
Δ	Difference between the matrix P and the matrix Q
γ_{uv}	Minimal traffic requirement between node u and node v
C	Maximal capacity of the network
c_i	Capacity of link i
$\vec{d}_i^{\,n}$	The ith column of the matrix Q_c^n
D_{ij}^c	Probability that the current capacity j of link i should be decreased
D_{uvlk}^p	Probability that the path l that carries k flows between node u and node v should be decreased
$\vec{e}_i^{\,n}$	The ith column of the matrix Q_{c+1}^n
f	Flow through a link
I_{ij}^c	Probability that the current capacity j of link i should be increased
I_{uvlk}^p	Probability that the path l that carries k flows between node u and node v should be increased
P_{ij}	Transition matrix from i units of flow to j units of flow
Q_{ij}	Same as P_{ij} with the exceptions that the first row, first column, last row and last column are set to zeroes
S_{ij}^c	Probability that the current capacity j of link i should be maintained
S_{uvlk}^p	Probability that the path l that carries k flows between node u and node v should be maintained

Distribution list

Document No.: DRDC CORA TM [enter number only: 9999-999]

LIST PART 1: Internal to Centre

1	Dr. Randy Martens (Email/pdf)
1	Dr. Alex Bourque (Email/pdf)
1	Mr. Mark Rempel (Email/pdf)
1	Dr. Dave Allen (Email/pdf)
1	Dr. Steve Guillouzic (Email/pdf)
1	Dr. Michel Couillard (Email/pdf)
1	Mr. Neil Carson (Email/pdf)
1	Mr. Peter De Jong (Email/pdf)
8	TOTAL LIST PART 1

LIST PART 2: External to Centre

1	DRDKIM (1EC)
1	DRDC Atlantic/Dr. Marcel LeFrançois (Email/pdf)
1	DRDC Atlantic/Mr. Vincent Myers (Email/pdf)
1	DRDC Atlantic/Dr. Mark Gammon (Email/pdf)
1	DRDC Atlantic/Mr. Mark Hazen (Email/pdf)
1	NURC/Dr. Ron Kessel (Email/pdf)
1	NC3A/Dr. Michael Crawford (Email/pdf)
7	TOTAL LIST PART 2

15 TOTAL COPIES REQUIRED

This page intentionally left blank.

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.) Defence R&D Canada – CORA 101 Colonel By Drive Ottawa, Ontario K1A 0K2	2. SECURITY CLASSIFICATION (Overall security classification of the document including special warning terms if applicable.) UNCLASSIFIED

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title.)

A Stochastic Optimization Algorithm using Intelligent Agents: With Constraints and Rate of Convergence

4. AUTHORS (last name, followed by initials – ranks, titles, etc. not to be used)

Bao U Nguyen

5. DATE OF PUBLICATION (Month and year of publication of document.) November 2010	6a. NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.) 32	6b. NO. OF REFS (Total cited in document.) 6

7. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Technical Memorandum

8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.)

Defence R&D Canada – CORA
101 Colonel By Drive
Ottawa, Ontario K1A 0K2

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)	9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)
10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.) DRDC CORA TM 2010-249	10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

11. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification.)

Unlimited

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.))

Unlimited

13. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

The problem of optimizing the average time latency of a network, using agents that are able to learn, is examined in this paper. The network design is constrained by a traffic matrix that dedicates specific flows between specific pairs of nodes. Although this is an application type of analysis, only the methodology is presented here, which includes an algorithm for optimization and a corresponding conservative rate of convergence based on no learning. The application part will be presented in the near future once data are available. It is expected that the tools developed in this paper can be used to optimize a wide range of objective functions that do not necessarily have to be the time latency. For example, it could be the cost of the network.

Le problème de l'optimisation du temps de latence moyen d'un réseau au moyen d'agents capables d'apprentissage, est examiné dans le présent document. La conception du réseau est contrainte par une matrice de trafic qui établit des flux particuliers entre des paires de nœuds particulières. Bien qu'il s'agisse d'un type de mise en application d'analyse, seulement les méthodologies sont présentées ici, y compris un algorithme d'optimisation et un taux de convergence correspondant raisonnable fondés sur un modèle sans apprentissage. La mise en application sera présentée prochainement, une fois les données disponibles. Il est espéré que les outils décrits dans le présent document permettront d'optimiser une vaste gamme de fonctions objectifs en plus de la latence.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Intelligent agents; optimization, Markov chain; rate of convergence, networks; time latencies; learning; matrices; induction.

Defence R&D Canada

Canada's Leader in Defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

DEFENCE DÉFENSE

www.drdc-rddc.gc.ca